

# Energy-Delay Tradeoffs in a Load-Balanced Router

Matthew Andrews  
Bell Labs, Murray Hill, NJ  
andrews@research.bell-labs.com

Lisa Zhang  
Bell Labs, Murray Hill, NJ  
ylz@research.bell-labs.com

**Abstract**—The Load-Balanced Router architecture has received a lot of attention because it does not require centralized scheduling at the internal switch fabrics. In this paper we reexamine the architecture, motivated by its potential to turn off multiple components and thereby conserve energy in the presence of low traffic.

We perform a detailed analysis of the queue and delay performance of a Load-Balanced Router under a simple random routing algorithm. We calculate probabilistic bounds for queue size and delay, and show that the probabilities drop exponentially with increasing queue size or delay. We also demonstrate a tradeoff in energy consumption against the queue and delay performance.

## I. INTRODUCTION

The concept of a *Load-Balanced Router* was studied at length at the beginning of last decade. See for example [6], [7], [22], [21], [9], [8], [15]. In this work we analyze various performance aspects of a Load-Balanced Router, motivated by the potential energy saving enabled by this architecture.

Energy efficiency in networking has recently attracted a large amount of attention. One of the main aims in much of this work is captured by the slogan *energy-follows-load*, also known as *energy-proportionality*. In other words, we wish to make sure that the energy consumed by a networking device matches the amount of traffic that the device needs to carry. This is in contrast to more traditional architectures for which the device operates at full rate at all times even if it is lightly loaded. Indeed, by a conservative estimate in a study conducted by the Department of Energy in 2008, at least 40% of the total consumption by network elements such as switches and routers can be saved if energy proportionality is achieved. This translates to a saving of 24 billion kWh per year attributed to data networking [1]. A recent study [12] further confirms that the power consumption of some state-of-the-art commercial routers stays within a small percentage of the peak power profile regardless of traffic fluctuation, for example the significant daily variation in traffic load [26].

Various approaches have been proposed in order to achieve energy-proportionality. *Speed scaling*, also known as *rate adaptation*, and *powering down* are two popular methods for effectively matching energy consumption to traffic load. The former refers to setting the processing speed of a network element according to traffic load. It is typically assumed that the energy consumption is superlinear with respect to the operating rate. The latter refers to turning off the element at certain times and so it either operates at the full rate or zero rate. Both methods are the subject of active research, though

most of the work focuses on optimizing an individual element in isolation [16], [14], [29], [23], [3], [5], [17], [18], [25], [13]. A central question to both methods is to set the speed so as to minimize energy usage while maintaining a desirable performance, e.g. latency or throughput.

The Load-Balanced Router architecture has the potential to handle the traffic in such a way that portions of the device can be turned off in response to lightly loaded traffic. We provide what we believe is the first detailed analysis of queue size and delay in a Load-Balanced Router, and we provide a tradeoff between energy consumption and queue size/delay. In order to describe our results in more detail we now give a brief description of the Load-Balanced Router architecture.

### A. Motivation for Traditional Load-Balanced Architecture

One of the most fundamental goals of any router architecture is to achieve *stability*, which is sometimes referred to as 100% throughput. In other words the router aims to process all the arriving traffic so long as no input and no output are inherently overloaded. The key difficulty with doing this is that the arriving traffic may be highly non-uniform, i.e. if  $A_{ik}$  is the arrival rate for traffic going from input  $i$  to output  $k$ , we will typically have  $A_{ik} \neq A_{i'k'}$  for  $ik \neq i'k'$ . Early work on switching considered a crossbar architecture in which matchings between the inputs and the outputs are set up at every time step. It was shown in [24] that the Maximum Weight Matching algorithm (with weights equal to the backlog for each input-output pair) can ensure stability. Subsequent papers looked at simplifications of this scheduler that could still achieve stability.

However, a major drawback of all these approaches is that they require a centralized scheduler with information about the backlogs of data on each input-output pair. A solution is the Load-Balanced Router that could make use of randomized routing ideas first proposed by Valiant [28]. In the Load-Balanced Router there is a middle stage placed between the input nodes and the output nodes. Each arriving packet is routed to a middle-stage node chosen at random. After passing through the middle stage each packet is then forwarded to its designated output. In order to realize this architecture we place a switching fabric between the input stage and middle stage and between the middle stage and the output stage. The beauty of this design is that the random routing ensures that for each of these switching fabrics no complicated scheduling is needed. All we need to do is repeat

a uniform schedule in which each connection is served at least once [6].

### B. Energy Consideration for Revisiting Load-Balanced Architecture

Our motivation for revisiting the Load-Balanced Router architecture is that it provides an attractive framework for studying energy proportionality [2]. For example, the number of active nodes in the middle stage can be reduced in the presence of light traffic, and increased with heavier traffic. The switch fabric between the input and middle stage and between the middle stage and the output can be functionally viewed as full meshes, as indicated in Figure 1. One possibility is to implement the mesh with round-robin crossbars. For example, Keslassy's thesis [22] assumes that the input, output and the middle stage are all of size  $n$  and each fabric is an  $n \times n$  crossbar that operates in a time-slotted fashion. At each time slot  $t$  the first fabric connects input node  $i$  to middle node  $(i+t) \bmod n$  and the second fabric connects middle node  $j$  to output node  $(j+t) \bmod n$ . In this implementation, if the number of active nodes in the middle stage is reduced then the crossbar can either slowdown or be turned off periodically.

Adjusting the size of the middle stage or the speed of the switching fabric requires considerable technological and engineering challenges. In this note we do not aim to address these issues. Our focus is on analyzing queue size and delay given the active portion of the middle stage, which leads to a tradeoff between power consumption and the size of the active middle stage.

### C. Model and Definition

We formally define the Load-Balanced Router architecture as follows. The router has  $n$  inputs and  $n$  outputs. We normalize the line rate such that it equals 1 on each input and output link. We also have a middle stage lying in between the inputs and outputs that consists of  $m$  nodes. The traditional Load-Balanced Router has  $m = n$ . However, here we treat  $m$  as a separate parameter. In between the input and the middle stage we have an  $n \times m$  mesh. Effectively each link in this mesh operates at rate  $\alpha/m$  for a *speedup* of  $\alpha \geq 1$ . Similarly, in between the middle and the output stage we have an  $m \times n$  mesh with link rate  $\beta/m$  for a speedup factor of  $\beta \geq 1$ . Each of the two meshes is *input-buffered*, i.e. each input node has a separate buffer for each middle node and each middle node has a separate buffer for each output node. See Figure 1.

From now on, we use  $i$  to index the input,  $j$  the middle stage and  $k$  the output. We refer to the packets that wish to go from input  $i$  to output  $k$  as  $ik$  packets. Similarly, link  $ij$  connects input  $i$  and node  $j$  in the middle stage and link  $jk$  connects node  $j$  in the middle stage and output  $k$ . Buffer  $ij$  (resp.  $jk$ ) at node  $i$  (resp.  $j$ ) buffers packets that are waiting to traverse link  $ij$  (resp.  $jk$ ).

The key to possible energy savings is that we assume that not all nodes need to be active at periods of low load. We suppose that at any time we can choose  $m' \leq m$  nodes to

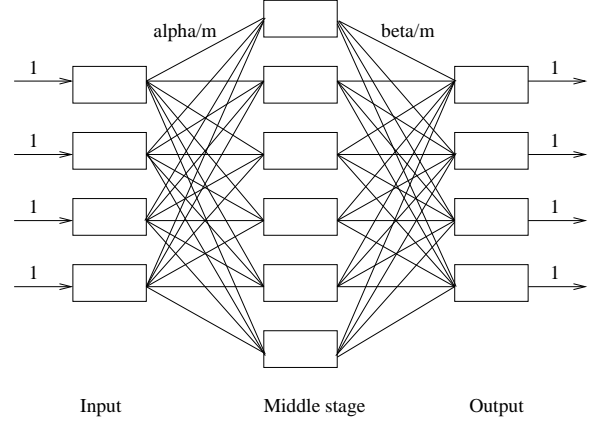


Fig. 1. A  $4 \times 6 \times 4$  Load-Balanced Router architecture, where the number of nodes in the middle stage can be different from the number of inputs and outputs.

be active in the middle stage and that such a configuration requires power  $w(m')$  for some function  $w(\cdot)$ .

When an  $ik$  packet  $p$  arrives we choose a random middle node  $j$  among the active nodes in the middle stage and place  $p$  in buffer at node  $ij$ .<sup>1</sup> The link  $ij$  operates continuously at rate  $\alpha/m$  and transmits packets in the  $ij$  buffer in a FIFO manner. Similarly, when  $p$  arrives at node  $j$  in the middle stage, it is placed in the buffer  $jk$ . The link  $jk$  continuously operates at rate  $\beta/m$  and transmits packets in the  $jk$  buffer in a FIFO manner. As we can see the scheduling for both stages requires no centralized intelligence and is extremely simple.

Lastly we describe our traffic model. As is common in work on scheduling in routers, we assume that packets are of unit size (or else are partitioned into cells of unit size). For any  $s, t$  let  $A_{ik}(s, t)$  be the amount of  $ik$  traffic arriving at the router in the time interval  $[s, t]$  and let  $A_i(s, t) = \sum_k A_{ik}(s, t)$  and  $A_k(s, t) = \sum_i A_{ik}(s, t)$ . We assume that the  $ik$  traffic, the input  $i$  traffic and the output  $k$  traffic are  $(\sigma_{ik}, r_{ik})$ ,  $(\sigma, 1)$  and  $(\sigma, 1 - \varepsilon)$  constrained respectively for some burst parameters  $\sigma_{ik}$  and  $\sigma$ , for some rate parameters  $r_{ik}$  and for some load parameter  $\varepsilon$ . In other words we assume that,

$$\begin{aligned} A_{ik}(s, t) &\leq \sigma_{ik} + r_{ik}(t - s) \\ A_i(s, t) &\leq \sigma + (t - s) \\ A_k(s, t) &\leq \sigma + (1 - \varepsilon)(t - s). \end{aligned}$$

We remark that the arrival rates  $r_{ik}$  will typically vary over time. Indeed, the energy savings that we hope to gain come precisely from the fact that we can match the number of active components to the traffic. However, we assume that this happens over a slow timescale and so we perform our scheduling analysis as if the rates  $r_{ik}$  are fixed.

<sup>1</sup>Note that round robin is another possibility for choosing a middle-stage node. However, a random choice is more robust against adversarial types of traffic arrivals. We do not go into details here.

#### D. Results

- In Section II we make the simple statement that  $\lceil \bar{r}m \rceil$  active middle-stage nodes suffice for handling the traffic load where  $\bar{r} \geq \sum_i r_{ik}$  for all  $k$  and  $\bar{r} \geq \sum_k r_{ik}$  for all  $i$ . This in turn implies that the energy required to serve the traffic over the long-term is  $w(\lceil \bar{r}m \rceil)$ .
  - In Sections III to IV-E we present a probabilistic analysis that bounds the queue sizes at the input and middle stages and bounds the delay experienced by packets as they travel from the router input to the router output. We first bound the probability for a queue size to exceed a certain amount  $q$ , and the delay to exceed a certain amount of  $d$ , assuming a fixed sized number of middle stages. An important feature of our bounds is that they decrease exponentially with  $q$  and  $d$ . For a fixed traffic load, we then derive a trade off between the queue size/delay performance and the number of active middle-stage nodes which in turn gives a energy-delay and energy-queue tradeoff.
- Leaving energy minimization aside, we believe that this is the first detailed analysis of the delay and queue performance of a Load-Balanced Router, which may be interesting in its own right.
- In Section V we present some numerical examples to validate our analytical findings.

We note that our approach is different from the traditional powerdown and rate adaptation techniques since we will be directing traffic in such a way that enables some components to be off. In other words for the middle stage nodes we are not trying to match service rate to a traffic process that is exogenous. We are trying to match the active middle stage nodes to a traffic process that is under our control due to our ability to route within the router.

We also remark that our bounds could be used to govern how many middle nodes are active in a Load-Balanced Router without necessarily computing all the bounds on the fly. We could instead precompute the bounds and create a simple look-up table that determines how many middle stage nodes should be active based on measurements of the load at the inputs and outputs.

## II. THROUGHPUT ANALYSIS

Recall that  $r_{ik}$  represents the current arrival rate of traffic that wishes to be routed from input  $i$  to output  $k$ . Let  $r_i = \sum_k r_{ik}$  and let  $r_k = \sum_i r_{ik}$ . Let  $\bar{r}$  be such that  $r_i \leq \bar{r}$  and  $r_k \leq \bar{r}$  for all  $i, k$ .

**Lemma 1.** *If we use  $m'$  middle stage elements then the router is not overloaded if  $m' \geq \bar{r}m$ . Hence the power required to serve all the traffic in the long-term is at most  $w(\lceil \bar{r}m \rceil)$ .*

*Proof:* Follows from the fact that if we turn on  $m'$  middle stage elements then for each middle element  $j$ ,  $1 \leq j \leq m'$ , the traffic rate that will be routed on link  $ij$  will be at most  $\bar{r}/m'$  which by assumption is at most  $(m'/m)/m' = 1/m$ . Since the capacity on the link  $ij$  is  $\alpha/m$

for some  $\alpha > 1$ , this implies that the  $ij$  link is not overloaded. A similar argument applies to each link  $jk$  between the middle and output stages  $\blacksquare$

## III. OVERVIEW OF DELAY ANALYSIS

Before diving into details of the delay analysis, we first provide a high level overview of our techniques.

### A. Relationship with Stochastic Network Calculus

We use a variant of network calculus [10], [11], [4] sometimes referred to as *stochastic network calculus*. The original form of network calculus derives delay bounds by imposing *upper bounds* on the amount of traffic arriving at a node via *arrival curves* and *lower bounds* on the amount of traffic served by a node via *service curves*. By relating these two curves we can both obtain a bound on the delay suffered by data at a network element and also characterize the arrival curves for the data at any downstream nodes. However, in the traditional network calculus all such bounds are required to hold with probability 1. In our context this will lead to extremely weak bounds since there is a non-zero probability that the router will send a large number of packets to a single middle-stage node, thus condemning them all to extremely poor service.

An alternative therefore is to use a *stochastic network calculus* in which we only wish for bounds on service to hold with high probability. A detailed formulation of a stochastic network calculus was outlined in a series of papers by Jiang and others [20], [19]. At a high level Jiang's approach obtains curves that bound the probability that the delay exceeds a certain amount at upstream elements, and then use these curves to bound the worst-case arrivals at downstream elements. However, we follow a slightly different approach since we are able to obtain better bounds by not directly utilizing the service curves at the input nodes to bound the arrivals at the middle-stage nodes. We instead base our calculations at the middle stage on the external arrivals of various ensembles of flows that might then be time-shifted due to delays at the input. This allows us to avoid handling complicated convolutions of arrival and service curves. We elaborate further on this distinction later.

### B. Our Approach

We divide our analysis into a series of pieces.

a) *Bound the queue build-up at the input:* Recall that between input  $i$  and middle-stage node  $j$  we effectively have a link with speed  $\alpha/m$ . Also recall that the input has a buffer especially dedicated to the traffic that wishes to go from input  $i$  to middle-stage node  $j$ . Suppose that at some time  $t$  this buffer has level  $q$ . Let  $s$  be the last time that this buffer was empty. Note that during the time interval  $[s, t)$  the  $ij$  link served data of total size  $\alpha(t-s)/m$ . Therefore the total data that arrived for link  $i$  during the time interval  $[s, t)$  is at least  $q + (\alpha(t-s)/m)$ .

Therefore the probability that link  $i$  has a backlog of size  $q$  at time  $t$  is upper bounded by the probability that for some

$s \leq t$ , the amount of  $ik$  data arriving at link  $i$  during the interval  $[s, t)$  is at least  $q + (\alpha(t - s)/m)$ . However, recall that the traffic arriving to input  $i$  arrives at rate at most  $\bar{r}$  and each packet is sent to a middle-stage node chosen uniformly at random. Hence, for fixed  $s$  and  $t$ , we can use a Chernoff bound to bound the probability that the amount of  $ik$  data arriving during the interval  $[s, t)$  is at least  $q + (\alpha(t - s)/m)$ . It is easy to show that this probability decreases exponentially in  $s$  and so we can use a union bound to bound the probability that this occurs for *any*  $s \leq t$ .

b) *Bound the delay experienced at the input:* Translating our bound on queue size into a bound on delay is simple. Since the transmission rate on the  $ij$  link is  $\alpha/m$ , the event that the head of line packet for the  $ij$  link at time  $t$  has experienced delay  $d$ , implies the event that at time  $t - d$ , the queue size for the  $ij$  link was at least  $d\alpha/m$ . An upper bound on the probability of the latter event can be derived using the method described earlier.

c) *Bound the queue build-up at middle stage:* We now give an overview of how we perform the analysis at the head of the  $jk$  queue at middle element  $j$ . This forms the crux of our analysis since we are in a more complicated situation due to the fact that the packet arrivals at the middle stage are affected by how they are served at the input. As before note that if the queue for link  $jk$  has size  $q$  at time  $t$ , then for some time  $s \leq t$ , the arrivals for link  $jk$  at middle-stage node  $j$  during the time interval  $[s, t)$  must be at least  $q + (\beta(t - s)/m)$ . Now suppose that the oldest of this data arrived at the input at time  $s - d$ , and suppose in addition that this data arrived at middle-stage node  $j$  on link  $ij$ . We can therefore state that the total amount of data arriving at the system in the time interval  $[s - d, t)$  that is destined for link  $jk$  is at least  $q + (\beta(t - s)/m)$  and the delay experienced by data arriving at link  $ik$  at time  $s - d$  is at least  $d$ . Via union bounds we can calculate the probability that this occurs for *any*  $i, s, d$ . In particular, for  $d$  small we can say that the probability that the arrivals exceed  $q + (\beta(t - s)/m)$  is small whereas if  $d$  is large then the probability that the link  $ik$  delay is at least  $d$  is small.

We make three points about this analysis at the middle stage.

- This is where our analysis deviates slightly from the traditional methodology of network calculus. We do not calculate a service curve for the input and use that service curve to bound the arrivals at the middle stage. Instead we analyze the delay behavior at the input and then use that calculation to bound the middle stage queue size using an expression that *still involves arrivals at the input*.
- Our initial analysis makes a slight approximating assumption in that for the  $ijk$  defined above, we treat the arriving traffic for link  $ij$  and the delay behavior on link  $ij$  as independent. In reality of course they will be correlated due to traffic that passes from input  $i$  to output  $k$  through middle-stage node  $j$ . However, this

approximation will typically not have a large effect for larger values of  $m$  since the  $ijk$  traffic only forms a  $1/m$  fraction of the total  $kj$ . However, in order to get a more accurate bound, in Section IV-E we present a more detailed expression that deals with the correlation explicitly.

- Our analysis is based on Chernoff bounds. However, the form of the Chernoff bound that we use changes depending on whether the bound on data arrivals that we are considering for a particular link is close to or far from the expected number of data arrivals for that link. This unfortunately leads to a somewhat involved case analysis.

d) *Bound the delay experienced at the middle stage:*

The conversion of the queue-size bound to a delay bound at the middle stage can be done in the exact same way as for the input. Now that we have an expression for the delay at both stages we can convert it into an expression for the end-to-end delay from the inputs to the outputs.

#### IV. ANALYTICAL BOUNDS ON DELAY

We now present the details of our delay analysis. We rely heavily on the following Chernoff bounds [27]. In particular we use (2) to derive analytical bounds and we use (1) to derive tighter bounds for numerical simulation.

**Theorem 2** (Chernoff Bound). *Let  $X_1, \dots, X_n$  be independent binary random variables, and let  $\mu$  be an upper bound on the expectation  $E[\sum_i X_i]$ . For all  $\delta > 0$ ,*

$$Pr[\sum_i X_i \geq (1 + \delta)\mu] \leq \left( \frac{e^\delta}{(1 + \delta)^{1 + \delta}} \right)^\mu \quad (1)$$

$$\leq e^{-\min(\delta^2, \delta) \cdot \mu / 3} \quad (2)$$

In what follows we sometimes refer to  $\mu$  as the *aggregated mean* and  $\delta$  as the *excess factor*. For conciseness we shall often use the aggregated mean even though  $\mu$  is strictly speaking a bound on the mean.

##### A. Input stage analysis

We begin by computing the queue distribution at the head of link  $ij$ , where  $i \in [1, n]$  is an input and  $j \in [1, m]$  is in the middle stage. As in Section II we assume that  $r_i \leq \bar{r}$  and  $r_k \leq \bar{r}$  for all  $i, k$  and we assume that  $m'$  middle stage elements are currently active. Initially, in order to keep the formulas manageable, we shall derive our formulas for the case in which  $\bar{r} = 1$  and  $m' = m$ . We shall also assume that the arriving traffic is smooth and so we do not have the burst terms  $\sigma_i$  and  $\sigma_k$ . Later on, we shall show how to adapt the formulas when these assumptions do not hold.

Let  $A_{ijk}(t)$  be the binary random variable that indicates whether a packet with input  $i$  output  $k$  is mapped to middle stage  $j$  at time  $t$ . We use  $A_{ijk}(t_1, t_2)$  to denote  $\sum_{t=t_1}^{t_2} A_{ijk}(t)$ , the total arrival in the duration  $(t_1, t_2]$ . Let  $Q_{i,j}^{(1)}(t)$  be the random variable for the queue size at the head of link  $ij$  at time  $t$ . To compute  $Q_{i,j}^{(1)}(t)$ , let us assume  $s < t$

be the last time that the queue at  $ij$  is empty. For  $Q_{ij}^{(1)}(t)$  to be larger than a value  $q$ , the arrival  $A_{ijk}(s, t)$  over all  $k \in [1, n]$  must be at least  $\alpha \cdot \frac{t-s}{m} + q$  since link  $ij$  is operated at a rate  $\frac{\alpha}{m}$ . Formally,

$$Pr[Q_{ij}^{(1)}(t) \geq q] \leq \sum_{s \leq t} Pr \left[ \sum_{k=1}^n A_{ijk}(s, t) \geq \alpha \cdot \frac{t-s}{m} + q \right] \quad (3)$$

We now bound the right-hand-side of (3). Since  $A_{ijk}(t)$  are independent binary variables, we apply the Chernoff bound to every term in the summation. The following is the aggregated mean  $\mu$  and the excess factor  $\delta$  for the above probability.

$$\begin{cases} \mu &= \frac{t-s}{m} \\ \delta &= \alpha - 1 + \frac{qm}{t-s} \end{cases},$$

since  $\alpha \cdot \frac{t-s}{m} + q = \mu(1 + \delta)$ . If  $\alpha \geq 2$ , we have  $\delta \geq 1$  and we use (2) to derive,

$$Pr[Q_{ij}^{(1)}(t) \geq q] \leq \sum_{t-s=0}^{\infty} e^{-\frac{t-s}{3m}(\alpha-1)-\frac{q}{3}} \leq \frac{e^{-\frac{q}{3}}}{1 - e^{-\frac{\alpha-1}{3m}}}. \quad (4)$$

If  $\alpha < 2$ , we have  $\delta \geq 1$  when  $t-s \leq \frac{qm}{2-\alpha}$ . Otherwise  $\delta < 1$ . We apply (2) as follows.

$$\begin{aligned} & Pr[Q_{ij}^{(1)}(t) \geq q] \\ & \leq \sum_{t-s=0}^{\frac{qm}{2-\alpha}} e^{-\frac{t-s}{3m}(\alpha-1)-\frac{q}{3}} + \sum_{t-s=\frac{qm}{2-\alpha}}^{\infty} e^{-\frac{t-s}{3m}(\alpha-1)^2} \\ & \leq \frac{e^{-\frac{q}{3}}}{1 - e^{-\frac{\alpha-1}{3m}}} + \frac{e^{-\frac{(\alpha-1)^2}{3(2-\alpha)}q}}{1 - e^{-\frac{(\alpha-1)^2}{3m}}} \end{aligned} \quad (5)$$

Note that both expressions are exponentially decreasing in  $q$ . For the time being we shall proceed according to the case  $\alpha \geq 2$  since this leads to more manageable formulas. Later on, we shall indicate where to adapt the formulas when we are in a scenario where  $\alpha < 2$ .

Let  $D_{ij}^{(1)}(t)$  be the maximum delay that some packet has experienced at time  $t$  in the queue  $Q_{ij}^{(1)}(t)$ . For this delay to be more than  $d$  at time  $t$ , some packet must be in the queue at time  $t-d$ . Since link  $ij$  operates at rate  $\frac{\alpha}{m}$  in a FIFO manner, the queue at time  $t-d$  must be at least  $d\alpha/m$ . Therefore,

$$Pr[D_{ij}^{(1)}(t) \geq d] \leq Pr[Q_{ij}^{(1)}(t-d) \geq d\alpha/m] \leq \frac{e^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}}.$$

By a union bound,

$$Pr[\exists i, D_{ij}^{(1)}(t) \geq d] \leq n \cdot \frac{e^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}}. \quad (6)$$

Recall that the above analysis was performed in the absence of the burst term  $\sigma$ . If we do have bursty traffic then we can adjust the formulas by making the following changes to the aggregated mean and the excess factor and propagating these changes through the resulting formulas.

$$\begin{cases} \mu &= \frac{t-s+\sigma}{m} \\ \delta &= \frac{(t-s)(\alpha-1)+qm-\sigma}{t-s+\sigma} \end{cases}$$

## B. Middle stage analysis

We now compute the queue distribution at the head of link  $jk$ , where  $j \in [1, m]$  is in the middle stage and  $k \in [1, n]$  is an output. Let  $Q_{jk}^{(2)}(t)$  be defined similarly as  $Q_{ij}^{(1)}(t)$ . To bound  $Q_{jk}^{(2)}$  at time  $t$ , let  $s \leq t$  be the last time that the queue  $Q_{jk}^{(2)}$  was empty. For  $Q_{jk}^{(2)}(t)$  to be larger than  $q$ , there must be at least  $\frac{(t-s)\beta}{m} + q$  distinct packets in  $Q_{jk}^{(2)}$  during the time period  $[s, t]$ . Further, let  $s-d$  be the earliest time one of these packets arrived at an input, say  $i$ . This packet must experience a delay of at least  $d$  in  $Q_{ij}^{(1)}$ . Therefore,

$$\begin{aligned} & Pr[Q_{jk}^{(2)}(t) \geq q] \\ & \leq \sum_d \sum_{s \leq t} Pr \left[ \sum_i A_{ijk}(s-d, t) \geq \frac{(t-s)\beta}{m} + q \right] \\ & \quad \cdot Pr[\exists i, D_{ij}^{(1)}(s) \geq d] \\ & \leq \sum_d \frac{ne^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}} \\ & \quad \cdot \sum_{s \leq t} Pr \left[ \sum_i A_{ijk}(s-d, t) \geq \frac{(t-s)\beta}{m} + q \right] \end{aligned}$$

Note that the bound (6) on the delay distribution at the input stage is independent of the time index. We can therefore move  $Pr[\exists i, D_{ij}^{(1)}(s) \geq d]$  to outside the summation indexed by time in the second inequality above.

We proceed to bound  $\sum_{s \leq t} Pr[\sum_i A_{ijk}(s-d, t) \geq \frac{(t-s)\beta}{m} + q]$  based on the following expressions for the expectation  $\mu$  and the excess factor  $\delta$ .

$$\begin{cases} \mu &= \frac{t-s+d}{m} \\ \delta &= \frac{(t-s)(\beta-1)+qm-d}{t-s+d} \end{cases},$$

since  $(1+\delta)\mu = \frac{(t-s)\beta}{m} + q$ . There are two cases to consider,  $\beta \leq 2$  or  $\beta > 2$ .

For  $\beta \leq 2$ , we further consider the following subcases, depending on  $\frac{qm}{d} - 1$ , the value of  $\delta$  when  $t-s=0$ . Note that as  $t-s$  increases, the value of  $\delta$  approaches  $\beta-1$ .

- *Case 1a:*  $1 \leq \frac{qm}{d} - 1$ , and  $t-s \leq \frac{qm-2d}{2-\beta}$ . In this case  $1 \leq \delta$ . Since  $\delta\mu = \frac{(t-s)(\beta-1)+qm-d}{m}$ , bound (2) implies (8).
- *Case 1b:*  $1 \leq \frac{qm}{d} - 1$ , and  $\frac{qm-2d}{2-\beta} \leq t-s$ . In this case  $\beta-1 \leq \delta \leq 1$ , which implies  $\delta^2\mu \geq (\beta-1)^2\mu$ . Bound (2) in turn implies (9).
- *Case 2:*  $\beta-1 \leq \frac{qm}{d} - 1 \leq 1$ , and for all values of  $t-s$ . In this case,  $\beta-1 \leq \delta \leq 1$ , which is the same situation as 1b and implies (10) in the same way.
- *Case 3a:*  $\frac{qm}{d} - 1 \leq \beta-1$ ,  $\frac{d(\beta+1)-2qm}{\beta-1} \leq t-s$  and  $\frac{d(\beta+1)-2qm}{\beta-1} \leq 0$ . In this case  $\frac{\beta-1}{2} \leq \delta \leq \beta-1$  for all values of  $t-s$ . Since  $\delta^2\mu \geq \frac{(\beta-1)^2\mu}{4}$ , bound (2) implies (11).

- *Case 3b:*  $\frac{qm}{d} - 1 \leq \beta - 1$ , and  $0 < \frac{d(\beta+1)-2qm}{\beta-1} \leq t-s$ . In this case  $\frac{\beta-1}{2} \leq \delta \leq \beta - 1$  for  $t-s \geq \frac{d(\beta+1)-2qm}{\beta-1}$ . Since  $\delta^2 \mu \geq \frac{(\beta-1)^2 \mu}{4}$ , bound (2) implies (12).
- *Case 3c:*  $\frac{qm}{d} - 1 \leq \beta - 1$ , and  $t-s < \frac{d(\beta+1)-2qm}{\beta-1}$ . We trivially upper bound the probability by 1 as in (13).

$$\begin{aligned} & Pr[Q_{jk}^{(2)}(t) \geq q] \\ & \leq \sum_d \sum_{s \leq t} Pr[\sum_i A_{ijk}^{(1)}(s-d, t) \geq \frac{(t-s)\beta}{m} + q] \quad (7) \\ & \quad \cdot Pr[\exists i, D_{ij}^{(1)}(s) \geq d] \end{aligned}$$

$$1a \leq \sum_{d=0}^{\frac{qm}{2}} \left( \frac{ne^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}} \right) \sum_{t-s=0}^{\frac{qm-2d}{2-\beta}} e^{-\frac{1}{3} \frac{(t-s)(\beta-1)+qm-d}{m}} + (8)$$

$$1b \sum_{d=0}^{\frac{qm}{2}} \left( \frac{ne^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}} \right) \sum_{t-s=\frac{qm-2d}{2-\beta}}^{\infty} e^{-\frac{1}{3}(\beta-1)^2 \frac{t-s+d}{m}} \quad (9)$$

$$2 \sum_{d=\frac{qm}{2}}^{\frac{qm}{\beta}} \left( \frac{ne^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}} \right) \sum_{t-s=0}^{\infty} e^{-\frac{1}{3}(\beta-1)^2 \frac{t-s+d}{m}} + (10)$$

$$3a \sum_{d=\frac{qm}{\beta}}^{\frac{2qm}{\beta+1}} \left( \frac{ne^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}} \right) \sum_{t-s=0}^{\infty} e^{-\frac{1}{3} \frac{(\beta-1)^2}{4} \frac{t-s+d}{m}} + (11)$$

$$3b \sum_{d=\frac{2qm}{\beta+1}}^{\infty} \left( \frac{ne^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}} \right)$$

$$\sum_{t-s=\frac{d(\beta+1)-2qm}{\beta-1}}^{\infty} e^{-\frac{1}{3} \frac{(\beta-1)^2}{4} \frac{t-s+d}{m}} + \quad (12)$$

$$3c \sum_{d=\frac{2qm}{\beta+1}}^{\infty} \left( \frac{ne^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}} \right) \sum_{t-s=0}^{\frac{d(\beta+1)-2qm}{\beta-1}} 1 \quad (13)$$

$$1a \leq \left( \frac{n}{1 - e^{-\frac{\alpha-1}{3m}}} \right) \left( \frac{e^{-\frac{q}{3}}}{1 - e^{-\frac{\beta-1}{3m}}} \right) \left( \frac{1}{1 - e^{-\frac{\alpha-1}{3m}}} \right) +$$

$$1b \left( \frac{n}{1 - e^{-\frac{\alpha-1}{3m}}} \right) \left( \frac{e^{-\frac{1}{3}(\beta-1)^2 q}}{1 - e^{-\frac{(\beta-1)^2}{3m}}} \right) \left( \frac{1}{1 - e^{-\frac{\alpha-(\beta-1)^2}{3m}}} \right)$$

$$2 \left( \frac{ne^{-\frac{\alpha}{3}q}}{1 - e^{-\frac{\alpha-1}{3m}}} \right) \left( \frac{e^{-\frac{1}{3}(\beta-1)^2 q}}{1 - e^{-\frac{(\beta-1)^2}{3m}}} \right) \left( \frac{1}{1 - e^{-\frac{\alpha-(\beta-1)^2}{3m}}} \right) +$$

$$3a \left( \frac{ne^{-\frac{\alpha}{3\beta}q}}{1 - e^{-\frac{\alpha-1}{3m}}} \right) \left( \frac{e^{-\frac{1}{12\beta}(\beta-1)^2 q}}{1 - e^{-\frac{(\beta-1)^2}{12m}}} \right) \left( \frac{1}{1 - e^{-\frac{4\alpha+(\beta-1)^2}{12m}}} \right) +$$

$$3b \left( \frac{ne^{-\frac{2\alpha}{3(\beta+1)}q}}{1 - e^{-\frac{\alpha-1}{3m}}} \right) \left( \frac{e^{-\frac{1}{6}(\beta-1)^2 q}}{1 - e^{-\frac{(\beta-1)^2}{12m}}} \right) \left( \frac{1}{1 - e^{-\frac{2\alpha+\beta(\beta-1)}{6m}}} \right) +$$

$$3c \left( \frac{(\beta+1)ne^{-\frac{\alpha}{3m}}}{(\beta-1)(1 - e^{-\frac{\alpha-1}{3m}})} \right) \left( \frac{1}{(1 - e^{-\frac{\alpha}{3m}})^2} \right)$$

$$\left( \left( \frac{2qm}{\beta+1} \right) e^{-\frac{\alpha}{3m} \left( \frac{2qm}{\beta+1} - 1 \right)} - \left( \frac{2qm}{\beta+1} - 1 \right) e^{-\frac{\alpha}{3m} \left( \frac{2qm}{\beta+1} \right)} \right)$$

Note that every term above decreases exponentially with the queue size  $q$ .

The case in which  $\beta > 2$  is simpler. We omit the analysis for space consideration. Recall that all of the above formulas are for the case  $\alpha \geq 2$ . If  $\alpha < 2$  then we need to replace all the factors (4),

$$\frac{e^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}}$$

with (5)

$$\frac{e^{-\frac{d\alpha}{3m}}}{1 - e^{-\frac{\alpha-1}{3m}}} + \frac{e^{-\frac{d\alpha(\alpha-1)^2}{3m(2-\alpha)}}}{1 - e^{-\frac{(\alpha-1)^2}{3m}}}.$$

If output  $k$  has a burst term  $\sigma$  then as in the input stage we can reflect this by adjusting the aggregated mean and the excess factor to,

$$\begin{cases} \mu &= \frac{t-s+d+\sigma}{m} \\ \delta &= \frac{(t-s)(\beta-1)+qm-d-\sigma}{t-s+d+\sigma} \end{cases}.$$

We conclude this section by bounding the delay distribution at the middle stage.  $D_{jk}^{(2)}$  be the maximum delay that some packet has experienced at time  $t$  in the queue  $Q_{jk}^{(2)}(t)$ . For this delay to be more than  $d$  at time  $t$ , some packet must be in the queue at time  $t-d$ . Since link  $jk$  operates in a FIFO manner, the queue at time  $t-d$  must be at least  $d\beta/m$ . Hence  $Pr[D_{jk}^{(2)}(t) \geq d] \leq Pr[Q_{jk}^{(2)}(t-d) \geq d\beta/m]$ .

We stress again that all of the above formulas are for the case  $\alpha \geq 2$ . If  $\alpha < 2$  then we need to replace all the factors (4) with factors (5).

### C. Eventual end-to-end delay

Now that we have delay bounds for the two stages of the router we can obtain a bound on the end-to-end delay distribution. In the following we let  $g_{m,\beta}(q)$  be a shorthand for the upper bound that we have derived on  $Pr[Q_{jk}^{(2)}(t) \geq q]$  and  $f_{m,\alpha}(q)$  a shorthand for our upper bound on  $Pr[Q_{ij}^{(1)}(t) \geq q]$ . Suppose that an  $ijk$  packet is still traversing the router at time  $t$ , but it arrived at the router before time  $t-d$ . It is not hard to see that either it is still waiting to traverse the input stage at time  $t - \frac{d}{2}$ , or it arrived at the middle stage by time  $t - \frac{d}{2}$ . Hence the probability that the end-to-end delay is at least  $d$  is at most,

$$Pr[D_{ij}^{(1)}(t - \frac{d}{2}) \geq \frac{d}{2}] + Pr[D_{jk}^{(2)}(t) \geq \frac{d}{2}]$$

$$\leq f_{m,\alpha}\left(\frac{d\alpha}{2m}\right) + g_{m,\beta}\left(\frac{d\beta}{2m}\right).$$

### D. Characterizing the tradeoff with the number of middle elements

In the above delay analysis we made a number of simplifying assumptions to keep the notation manageable. For example we held  $m' = m$  and  $\bar{r} = 1$ . We now demonstrate that we can use the above formulas to handle the case of arbitrary  $m'$  and  $\bar{r}$ . This in turn allows to characterize the tradeoff between energy consumption and end-to-end delay.

The main idea is to scale time so that the arrival rate at the inputs is scaled to 1. We also adjust the link rates on the two stages of the mesh. In particular, if we wish to analyze a system with a given  $m, m', \bar{r}, \alpha$  and  $\beta$ , we define a new system characterized by  $\hat{m}, \hat{m}', \hat{r}, \hat{\alpha}$  and  $\hat{\beta}$  in which we set,

$$\hat{m}' = \hat{m} = m \quad \hat{r} = 1 \quad \hat{\alpha} = \frac{\alpha m'}{m \bar{r}} \quad \hat{\beta} = \frac{\beta m'}{m \bar{r}}$$

Then it is not hard to see that if we scale time by a factor  $\bar{r}$ , the new system has exactly the same behavior as the old one. However, we are now working in a system with  $\hat{m}' = \hat{m}$  and  $\hat{r} = 1$ . Hence we can apply the analysis that we have already derived.

Our main result is thus,

**Theorem 3.** *If we run the router with  $m'$  middle stage elements then it uses energy  $w(m')$  and the probability that the end-to-end delay is at least  $d$  is bounded by,*

$$f_{m', \frac{\alpha m'}{m \bar{r}}}(\frac{d\alpha}{2m\bar{r}}) + g_{m', \frac{\beta m'}{m \bar{r}}}(\frac{d\beta}{2m\bar{r}}).$$

#### E. Dealing with dependence

In our analysis of the middle stage we implicitly made the simplifying assumption that  $\sum_{i'} A_{i'jk}^{(1)}(s-d, t)$  is independent from  $D_{ij}(s)$ . However, this is not strictly true since the arrivals for the path  $ijk$  will affect both  $\sum_{i'} A_{i'jk}^{(1)}(s-d, t)$  as well as  $D_{ij}(s)$ . Since the  $ijk$  flow represents only a  $1/m$  fraction of the traffic that contributes to  $D_{ij}(s)$ , this will typically have a negligible effect on the eventual results. However, if we required a true upper bound on the probability distribution of  $Q_{jk}^{(2)}$  we must use the following adaptation of the formula, which for each  $i$ , conditions the event  $D_{ij}^{(1)}(s) \geq d$  on whether or not  $A_{ijk}^{(1)}(s-d, t)$  is greater than  $\frac{5(t-s)\beta}{mn}$ . More formally,

$$\begin{aligned} & Pr[Q_{jk}^{(2)}(t) \geq q] \\ & \leq \sum_d \sum_{s \leq t} \left( \sum_i Pr[A_{ijk}^{(1)}(s-d, t) \geq \frac{5(t-s)\beta}{mn}] + \right. \\ & \quad \left. \sum_i Pr[\sum_{i \neq i} A_{ijk}^{(1)}(s-d, t) \geq \frac{(t-s)\beta}{m} + q - \frac{5(t-s)\beta}{mn}] \right. \\ & \quad \left. Pr[D_{ij}^{(1)}(s) \geq d | A_{ijk}^{(1)}(s-d, t) \leq \frac{5(t-s)\beta}{mn}] \right) \end{aligned}$$

#### F. Queues at output

We conclude this section by explaining how the analysis can be extended if we also wish to bound the delay on the output link from the router. Let  $Q_k^{(3)}(t)$  be the queue at the head of the output link and recall that this link has speed 1. To bound  $Q_k^{(3)}(t)$  at time  $t$ , let  $s \leq t$  be the last time that queue  $Q_k^{(3)}$  was empty. For  $Q_k^{(3)}$  to be larger than  $q$ , there must be at least  $t-s+q$  distinct packets in  $Q_k^{(3)}$  during the time period  $[s, t]$ . Further, let  $s-d$  be the earliest time one of these packets arrived at an input. Suppose that the path of this packet is  $ijk$ . Then the packet must either be waiting in

the queue  $Q_{ij}^{(1)}$  at time  $s-d$  or it must be waiting in the queue  $Q_{jk}^{(2)}$  at time  $s-d$ . In the former case the packet must have experienced delay of  $\frac{d}{2}$  in  $Q_{ij}^{(1)}$  at time  $s-d$  and in the latter case it must have experienced delay of  $\frac{d}{2}$  in  $Q_{jk}^{(2)}$  at time  $s$ . Therefore,

$$\begin{aligned} & Pr[Q_k^{(3)}(t) \geq q] \\ & \leq \sum_d \sum_{s \leq t} Pr \left[ \sum_{ij} A_{ijk}(s-d, t) \geq t-s+q \right] \\ & \quad \left( Pr[\exists ij, D_{ij}^{(1)}(s-d) \geq \frac{d}{2}] + Pr[\exists j, D_{jk}^{(2)}(s) \geq \frac{d}{2}] \right) \\ & \leq \sum_d nm f_{m, \alpha}(\frac{d\alpha}{2m}) \cdot mg_{m, \beta}(\frac{d\beta}{2m}) \cdot \frac{1}{\varepsilon} (d-q+\sigma_k). \end{aligned}$$

In addition, as before, we can convert this bound to a bound on delay for the third stage. Let  $D_k^{(3)}$  be the maximum delay that some packet has experienced at time  $t$  in the queue  $Q_k^{(3)}(t)$ . For this delay to be more than  $d$  at time  $t$ , some packet must be in the queue at time  $t-d$ . Since link  $k$  operates in a FIFO manner, the queue at time  $t-d$  must be at least  $d(1-\varepsilon)$ . Hence  $Pr[D_k^{(3)}(t) \geq d] \leq Pr[Q_k^{(3)}(t-d) \geq d(1-\varepsilon)] \leq \sum_{d'} nm f_{m, \alpha}(\frac{d'\alpha}{2m}) \cdot mg_{m, \beta}(\frac{d'\beta}{2m}) \cdot \frac{1}{\varepsilon} (d'-d(1-\varepsilon)+\sigma_k)$ .

### V. NUMERICAL RESULTS

In this section we show numerical examples of the queue bounds. In particular for these calculations we use formulas that are similar to those derived in Section IV but we use Chernoff bounds of the form (1) rather than (2) since the former are tighter.

Figure 2 plots the logarithm of the probability of a middle-stage queue  $jk$  exceeding  $q$  packets against the queue size  $q$ . For this instance, the router is  $20 \times 80 \times 20$ . The traffic is fully loaded for each of the inputs and outputs. We vary the speedup  $\alpha = \beta$  in the range of 2, 3, 4 and 5. As we can see from the figure the logarithm of the probability decreases linearly with the queue size, which means the probability decreases exponentially with the queue. As expected, we can also see that with increasing speedup, the probability of the queue size exceeding  $q$  drops.

Figure 3 demonstrates the tradeoff between the queue size and the number of active middle-stage nodes. For this instance, the router is again  $20 \times 80 \times 20$ . The link rate of the interconnect is set to  $1/20$ . The traffic is fully loaded for each of the inputs and outputs. If we keep all  $m = 80$  nodes in the middle stage active, we can see from the bottom curve of Figure 3 that the queue size is the smallest. However, this option is also the most energy consuming as all 80 middle-stage nodes are kept active. For the other extreme, we can activate 40 middle-stage nodes, which is the most energy efficient. However, we can see from the bottom curve of Figure 3 that the queue size is the smallest. The curves in between correspond to the intermediate cases in which the number of active middle-stage nodes are 50, 60 and 70.

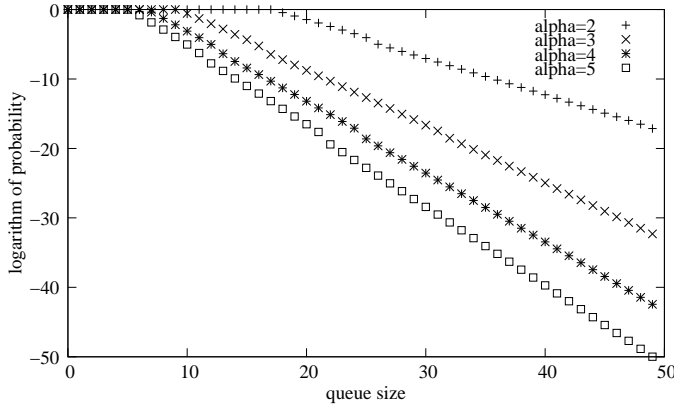


Fig. 2. Log of probability against queue size, for  $n = 20$ ,  $m = 80$  and fully loaded traffic. From top to bottom, the curves correspond to increasing speedup from  $\alpha = \beta = 2, 3, 4$  and  $5$ .

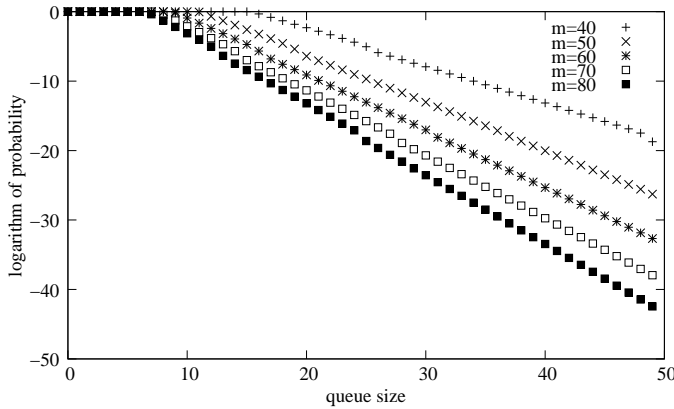


Fig. 3. Log of probability against queue size, for same amount of total traffic but varying number of active middle-stage nodes. From top to bottom, the curves correspond to increasing number of active middle-stage nodes from  $m = 40, 50, 60, 70, 80$ . The number of inputs and outputs is  $n = 20$  and interconnect link rate is  $1/20$ .

## VI. CONCLUSION

In this paper we revisit the Load-Balanced Router architecture, motivated by its potential of delivering energy proportionality for routers. We offer a detailed analysis on the queue lengths and packet delays under a simple random routing algorithm which is robust against all admissible traffic. This allows us to observe a trade off between performances such as queue size against energy consumption.

Our paper does not focus on algorithms that optimize the number of active middle-stage nodes. We give a very simple argument for which the size of the active middle stage is proportional to the *maximum* traffic load over all inputs and outputs. It is an intriguing open question to see how one could make sure that the size of the active middle stage is proportional to the traffic average over all input, not to the maximum.

## REFERENCES

- [1] Routing telecom and data centers toward efficient energy use. In *Proceedings of the Vision and Roadmap Workshop, U.S. Department of Energy*, October 2008.
- [2] S. Antonakopoulos, S. Fortune, A. Francini, T. Klein, R. McLellan, D. Nielson, and L. Zhang. Personal communication, 2011.

- [3] N. Bansal, T. Kimbrel, and K. Pruhs. Speed scaling to manage energy and temperature. *Journal of the ACM*, 54(1), 2007.
- [4] J. Y. L. Boudec and P. Thiran. *Network Calculus*. Springer Verlag, [http://ica1www.epfl.ch/PS\\_files/NetCal.htm](http://ica1www.epfl.ch/PS_files/NetCal.htm), 2004.
- [5] H.-L. Chan, W.-T. Chan, T. W. Lam, L.-K. Lee, K.-S. Mak, and P. W. H. Wong. Energy efficient online deadline scheduling. In *Proceedings of ACM-SIAM SODA*, pages 795–804, 2007.
- [6] C. Chang, D. Lee, and Y. Jou. Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering. In *IEEE HPSR '01*, Dallas, TX, 2001.
- [7] C. Chang, D. Lee, and C. Lien. Load balanced Birkhoff-von Neumann switches, part II: multi-stage buffering. *Computer Comm.*, 25:623 – 634, 2002.
- [8] C. Chang, D. Lee, and Y. Shih. Mailbox switch: a scalable two-stage switch architecture for conflict resolution of ordered packets. In *Proceedings of IEEE INFOCOM 2004*, Hong Kong, 2004.
- [9] C. Chang, D. Lee, and C. Yue. Providing guaranteed rate services in the load balanced Birkhoff-von Neumann switches. In *Proceedings of IEEE INFOCOM 2003*, San Francisco, CA, 2003.
- [10] R. L. Cruz. A calculus for network delay, Part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114 – 131, 1991.
- [11] R. L. Cruz. A calculus for network delay, Part II: Network analysis. *IEEE Transactions on Information Theory*, 37(1):132 – 141, 1991.
- [12] A. Francini, S. Fortune, T. Klein, and M. Ricca. Energy profiling of network equipment for rate adaptation technologies. Internal Technical Documents, Alcatel-Lucent, 2011.
- [13] A. Francini and D. Stiliadis. Energy efficiency with rate adaptation. In *Proc. of IEEE HPSR*, 2010. <http://ect.bell-labs.com/who/francini/ra9.pdf>.
- [14] M. Garrett. Powering down. *Commun. ACM*, 51(9):42–46, 2008.
- [15] N. M. I. Keslassy, S.T. Chuang. A load-balanced switch with an arbitrary number of linecards. In *Proceedings of IEEE INFOCOM 2004*, Hong Kong, 2004.
- [16] S. Irani and K. Pruhs. Algorithmic problems in power management. *SIGACT News*, 36(2):63–76, 2005.
- [17] S. Irani, S. K. Shukla, and R. Gupta. Algorithms for power savings. *ACM Transactions on Algorithms*, 3(4), 2007.
- [18] S. Irani, S. K. Shukla, and R. K. Gupta. Online strategies for dynamic power management in systems with multiple power-saving states. *ACM Trans. Embedded Comput. Syst.*, 2(3):325–346, 2003.
- [19] Y. Jiang. A basic stochastic network calculus. In *Proceedings of ACM SIGCOMM '06*, pages 123 – 134, New York, NY, 2006.
- [20] Y. Jiang and P. J. Emstad. Analysis of stochastic service guarantees in communication networks: A server model. In *In Proc. of the International Workshop on Quality of Service (IWQoS 2005)*, pages 233–245, 2005.
- [21] I. Keslassy, S.-T. Chuang, D. M. K. Yu, M. Horowitz, and O. Solgaard. Scaling internet routers using optics. In *Proceedings of ACM SIGCOMM '03*, Karlsruhe, Germany, 2003.
- [22] I. Keslassy and N. McKeown. Maintaining packet order in two-stage switches. In *Proceedings of IEEE INFOCOM 2002*, New York, NY, June 2002.
- [23] M. Li, B. J. Liu, and F. F. Yao. Min-energy voltage allocation for tree-structured tasks. In *Proceedings of COCOON*, pages 283–296, 2005.
- [24] N. W. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. In *Proceedings of IEEE INFOCOM '96*, pages 296 – 302, San Francisco, CA, March 1996.
- [25] S. Nedeveschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing network energy consumption via sleeping and rate-adaptation. In *Proceedings of NSDI*, pages 323–336, 2008.
- [26] M. Roughan, A. Greenber, C. Kalmanek, M. Rumeszczewicz, J. Yates, and Y. Zhang. Experience in measuring internet backbone traffic variability. In *Proc. ACM SIGCOMM IMW*, pages 91 – 92, 2002.
- [27] C. Scheidler. *Probabilistic Methods for Coordination Problems*. Habilitation thesis, Paderborn University, 2000.
- [28] L. G. Valiant. A scheme for fast parallel communication. *SIAM J. Comput.*, 11(2):350 – 361, 1982.
- [29] F. F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced CPU energy. In *Proceedings of IEEE FOCS*, pages 374–382, 1995.